**1 引言**

**1.1 编写目的**

本说明书目的在于明确说明系统各功能的实现方式，指导开发员进行编码。

本说明书的预期读者为：系统设计者、系统开发员。

**1.2 背景**

待开发网页的名称： 游戏论坛

此软件系统任务提出者：客户（游戏爱好者）

此软件系统任务开发者：第七小组

此软件系统任务用户：需要提供用户浏览相关游戏资讯，论坛发表评论，发布帖子等功能。

**2 总体设计**

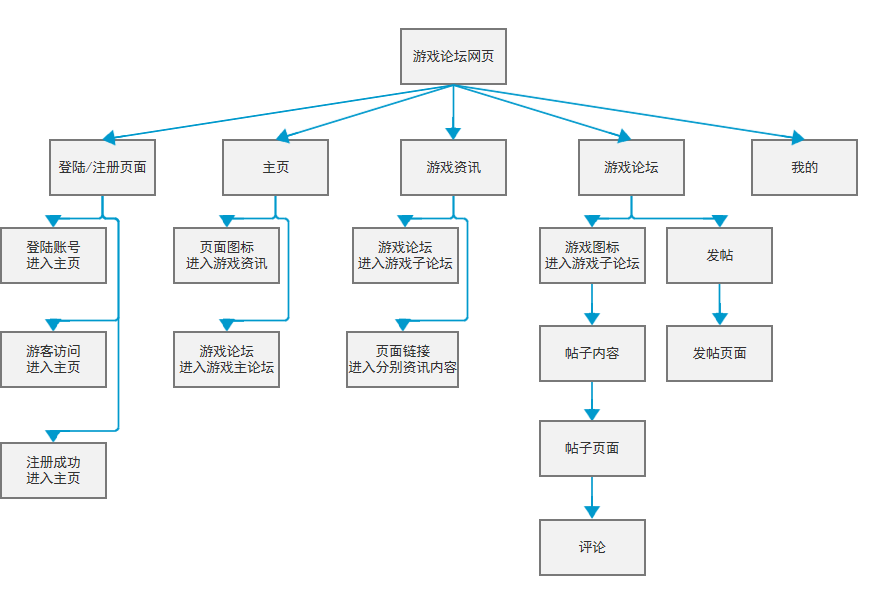
**2.1 概述**

**2.3 系统总体结构设计**

**2.3.1 系统业务层次图**

内容描述：游戏论坛网页是服务于游戏爱好者的页面，旨在为用户提供一个了解游戏资讯，发表评论与帖子的交流平台，具有信息共享，资源合理分配，业务协调统一的特点。它分为登陆/注册面，主页，游戏资讯页面，论坛页面，我的页面等五大主题页面。

 图例：



功能简介：

登陆/注册页面：账号登陆，账号注册，也可以选择游客访问。

主页面： 主页面上有导航栏，可以进入各个主题页面，通过导航栏直接链接到游戏主论坛页面。点击游戏区域的游戏主题可以进入分别的游戏资讯页面，还可以选择二维码加入公众号。

游戏资讯页面：点击游戏论坛可以进入分别的游戏论坛子页面，点击页面中的游戏攻略，赛事资讯等页面内容可以进行游戏各方面资讯的浏览。

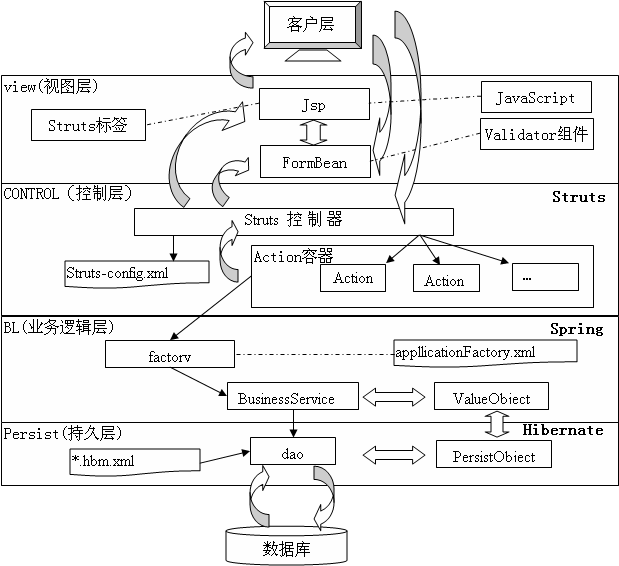
游戏论坛页面：选择页面中的不同游戏可以分别进入不同的游戏论坛子页面。

游戏论坛子页面：可以选择发帖进入发帖页面，点击页面中排列的帖子内容进入帖子界面。

发帖界面：编辑好自己想发表的内容，点击提交，提交至论坛子页面。

帖子界面：观看帖子内容和网友评论，页面底部可以发表自己的评论。

我的界面：了解自己的相关信息，如兴趣部落，我的关注，我的帖子等信息。



SSH架构结构图（图例1）

n 各层实现功能及开发技术说明

1.四层结构的优势

1）通过成熟的开源产品实现各层功能开发，比起自己开发能缩短开发周期，且架构所用到的开源产品均有很广泛的用户群，经受过实践的考验，质量和性能更有保障。

2）层与层之间松散偶合，增加代码重用率。

    3)各层分工明确，这样也利于团队的明确分工。

2.表示层

这一层是面向用户的界面，是用户与系统之间交互的媒介。如：用户在界面发送请求，系统接收请求，进行处理，然后通过界面将结果呈现于用户。这一过程包括了用户动作、数据传递、界面显示。大家熟悉的MVC模式就是将这三者分离，减少三者耦合。我们在该层借助了Struts来实现。

2）Struts的实现的功能:

管理用户的请求，做出相应的响应。

提供一个Controller，委派调用业务逻辑和其它上层处理。

处理异常，抛给Struts Action.

为显示提供一个模型。

UI（User interface）验证。

以下部分则不该在Struts显示层的编码中经常出现。因为在表示层引入这些代码，则会带来高偶合和非常麻烦的维护代价。

直接的与数据库通信，如JDBC调用。

与你应用程序相关联的业务逻辑以及校验。

3.业务层

业务层在实际的项目开发中，每个领域都会有自己独特的业务逻辑，正因为这样，致使项目中代码高度偶合，原本有可能被重用的代码或功能，因为与具体的业务逻辑绑定在一块而导致很难被重用。因此我们将实现这些具体逻辑的代码抽取出来分为单独的一层，其目的是希望通过分层，来降低它与系统其他部分的偶合度。

现实中世界是变化的，既然该层实现的是现实中具体的业务逻辑，那该层的实现代码不可避免的会发生变更。怎样让该层适应最大的变化，做到最小的改动?通常我们在编码的时候会尽量考虑到同一业务多种实现的兼容和可扩展的能力。因此我们在该层借助了Spring，通过依赖注入、AOP应用、面向接口编程，来降低业务组件之间的偶合度，增强系统扩展性。

2）Spring实现的功能:

处理应用程序的业务逻辑和业务校验。

管理事务。

提供与其它层协同工作的接口。

管理业务层级别的对象的依赖。

在显示层和持久层之间增加了一个灵活的机制，使得他们不直接的联系在一起。

通过揭示从显示层到业务层之间的Context来得到business services。

管理程序的执行（从业务层到持久层)。

4.数据持久层

    数据持久层在开发中与数据库进行数据交互必不可少，通常我们归为CRUD（添加、读取、修改、删除)，这些操作占据了系统开发中大部分的时间，同时我们还需要考虑与数据库交互的性能问题，如连接池、数据缓存等等。系统内部的持续层不但需要大量调试时间，而且还经常缺少功能使之变得难以控制。针对这点我们引入ORM开源架构Hibernate。

2）Hibernate实现的功能:

查询对象的相关信息的语句。

存储，更新，删除数据库记录。

支持大部分主流数据库，并且支持Parent/Child关系，事物处理，继承和多态。

5.各层中的封装类

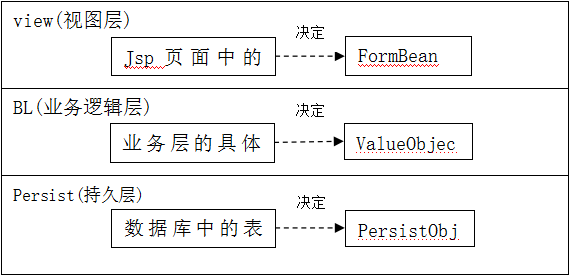
各层的封装类的主要功能都是一致的，就是将有一定联系的数据集合装载在其实力对象中，这样做的道路是显而易见的，通过对象来传递数据集合的效率会更高更方便。

显示层的FormBean类是用封装来自页面form提交的信息的，一般情况下这个类的私有变量是与页面form的元素一一对应的。另外FormBean封装类的对象还负责将要显示的信息传递到显示层，其作用是双向的。

业务逻辑层的ValueObject类是用来封装一定业务功能实现过程中需要的数据集合的，也就是说要封装的数据都是由业务功能的需要决定的。

持久层的PersistObject类，其实例化对象所封装的数据集是与数据库中表相对应的，即表项对应要封装的数据项。

我们根据上面封装类的说明可以看出，FormBean、ValueObject、PersistObject、三者的作用相似都是为了封装数据信息，不同的是这些对象所在的是不同架构层面，这样做的好处是数据的处理和转递比较有条理，层次清晰易于维护。封装类在架构中的情况如图4.4所示。



各层封装类的情况（图例2）

**2.3.3 关键技术与算法(选作)**

书写要求：

对系统中比较复杂的功能的实现方式或应用的特殊技术加以说明要求通俗直观便于客户参考。

书写样例：

查询统计（科学项目的查询和统计，科学成果的查询和统计，科学奖励的查询和统计，科技人才的查询和统计）

**组合检索**：组合检索由拥有搜索权限的用户在组合类别的下拉框中选择需要查询的项目，如科技项目的检索：项目类别+项目资金+项目所属公司+时间；没有选择的项，下拉框默认为空白。点击搜索，系统将对数据库的字段进行模糊搜索，搜索结果列表显示于组合搜索下面，分页显示搜索结果。

*注：其它的科技成果搜索，科技奖励搜索，科技人才搜索采用统一的组合搜索形式*

**统计图表**：各分公司统计：根据用户在科技项目申请，科技成果申请，科技奖励申请，科技人才的输入的数据，当拥有该权限的用户点击各分公司统计，系统将从数据库中读取以上的部分数据，进行统计，然后以统计图形的形式显示出来，对各分公司进行比较，方便领导制定企业决策。

**某分公司统计**：当拥有权限的用户点击某分公司统计，系统将从数据库中读取这对该公司的数据进行统计，显示统计图，该统计主要面向集团和分公司领导。

*注：其它的科技成果搜索，科技奖励搜索，科技人才搜索采用统一的统计形式*

**2.2.4关键数据结构（数据库设计）**

书写要求：

简要说明本系统中的最主要的数据结构。

书写样例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 允许空 | 默认值 | 约束 | 备注 |
| CODE | VARCHAR2(3) | N |  | 主键 | 编码 |
| NAME | VARCHAR2(10) | N |  |  |  |
| SEX | VARCHAR2(3) | N | male |  |  |

列名 数据类型 允许空 默认值 约束 备注

**3 系统功能设计**

**3.1 论坛大体功能**

3.1.1 **功能描述**

帖子的编写可以通过发帖页面进行编辑，点击提交通过后台将内容上传至论坛子页面，然后将数据保持至数据库中显示至论坛子页面。如图3.1.1\_1。



  图3.1.1\_1

发表评论在帖子页面下方进行，点击提交按钮即可发表自己的评论。如图3.1.1\_2

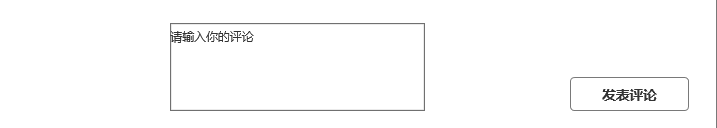
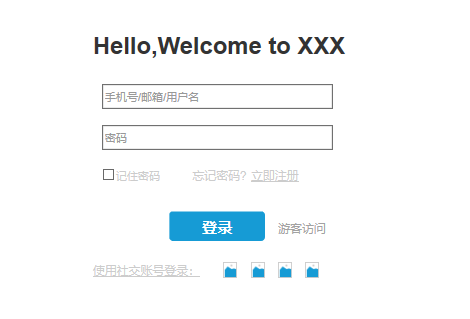


图3.1.1\_2

 登陆界面输入账号密码，即可登陆账号，进入主界面。



若无账号，点击注册页面进行注册，跳转至主页面。

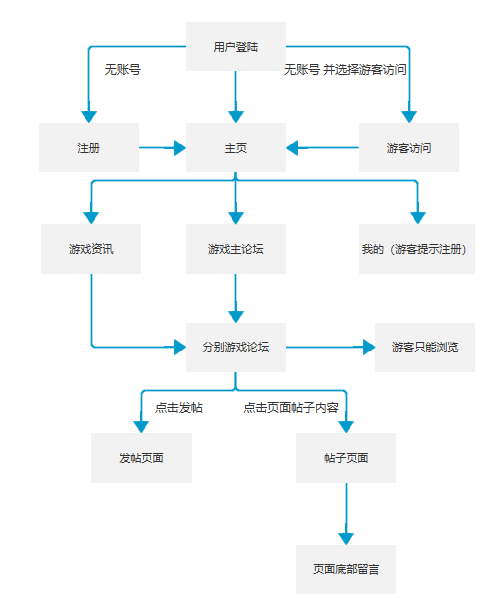
3.1.2 **接口设计**

输入操作：客户或管理员输入订单详细信息并提交。

输出效果：订单成功进入待审列表。

管理员对待审列表中的订单进行审核，填写相关审核结果。

3.1.3 **功能流程图**



3.2.2 **接口设计**

输入操作：查询并选择待统计订单，计算月收益。

输出效果：得到每月收益。

输入操作通过界面进行。

输入操作：输入一时间段。

输出效果：显示该时间段内汇报的统计列表。

输入操作通过界面进行。

**4 尚待解决的问题**

暂无...